```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score


Hipótese: "Músicas com BPM (Beats Per Minute) mais altos fazem mais sucesso em termos de streams no Spotify"


df = pd.read_csv('tabelas_unificadas.csv')  # Substitua pelo nome do seu arquivo
df.head()
```

	track_id	artist_count	in_spotify_playlists	in_spotify_charts	track_corrigida	artista_corrigida	streams_li
0	3210546	1	2468	0	Selfish	PnB Rock	380319
1	3910891	1	3995	13	Just Wanna Rock	Lil Uzi Vert	457184
2	4635311	1	1776	14	After Dark	MrKitty	646886
3	6327262	1	7556	0	Falling	Harry Styles	1023187
4	5046895	1	7109	2	Talking To The Moon	Bruno Mars	1062956

5 rows × 43 columns

```
print(df.columns)
```

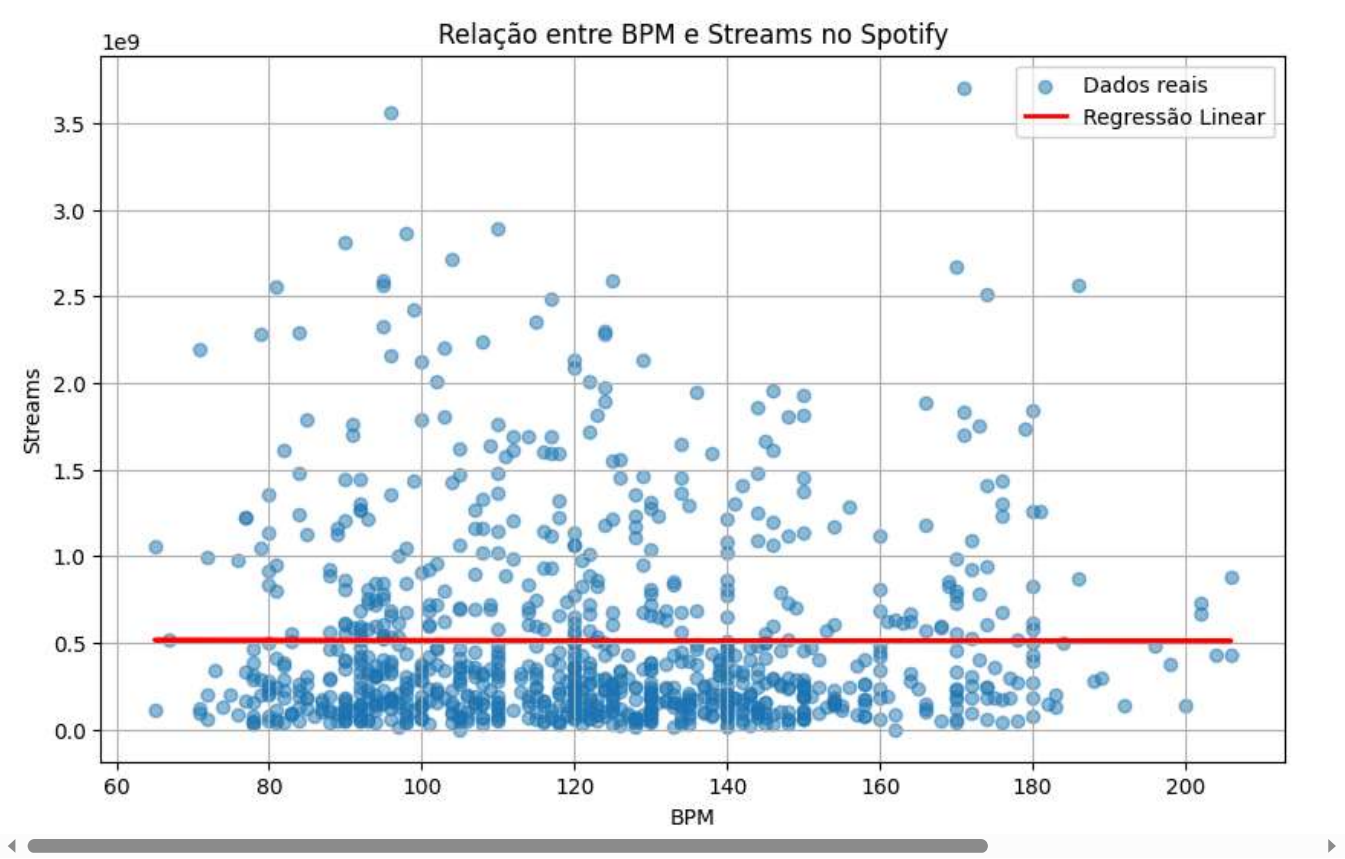
```
Index(['track_id', 'artist_count', 'in_spotify_playlists', 'in_spotify_charts',
       'track_corrigida', 'artista_corrigida', 'streams_limpo',
       'data_lancamento', 'in_apple_charts', 'in_apple_playlists',
       'in_deezer_charts', 'in_deezer_playlists', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'acousticness_%', 'danceability_%', 'energy_%',
       'instrumentalness_%', 'liveness_%', 'speechiness_%', 'valence_%',
       'total_playlists', 'charts_concorrentes', 'playlist_quartil',
       'playlist_categorizada', 'bpm_quartil', 'bpm_categorizada',
       'danceability_quartil', 'danceability_categorizada', 'valence_quartil',
       'valence_categorizada', 'energy_quartil', 'energy_categorizada',
       'acousticness_quartil', 'acousticness_categorizada',
       'instrumentalness_quartil', 'instrumentalness_categorizada',
       'liveness_quartil', 'liveness_categorizada', 'speechiness_quartil',
       'speechiness_categorizada'],
      dtype='object')
```

```
X = df[['bpm']]  # variável independente
y = df['streams_limpo']  # variável dependente


modelo = LinearRegression()
modelo.fit(X, y)

# Previsões
y_pred = modelo.predict(X)


print("Inclinação (coeficiente):", modelo.coef_[0])
print("Intercepto:", modelo.intercept_)
print("R² (coeficiente de determinação):", r2_score(y, y_pred))
```

```
Inclinação (coeficiente): -45282.17883646517
Intercepto: 519955118.2436913
R² (coeficiente de determinação): 5.00462120267553e-06
```

```
plt.figure(figsize=(10,6))
plt.scatter(X, y, alpha=0.5, label='Dados reais')
plt.plot(X, y_pred, color='red', linewidth=2, label='Regressão Linear')
plt.title('Relação entre BPM e Streams no Spotify')
plt.xlabel('BPM')
plt.ylabel('Streams')
plt.legend()
```

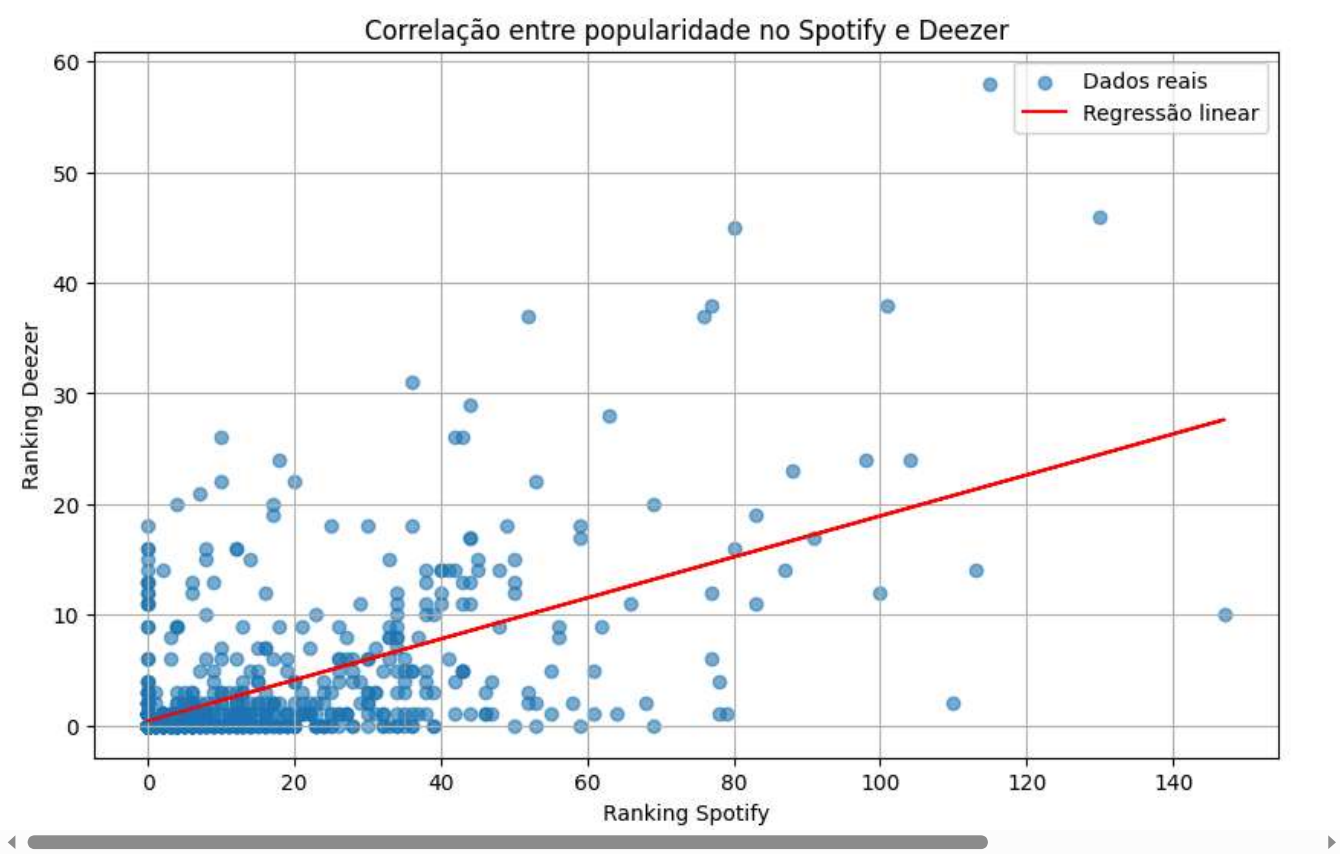
```
plt.grid(True)
plt.show()
```




```
# Resultado: hipótese indeterminada, não é possível identificar relação entre as variáveis.


# Hipótese: "As músicas mais populares no ranking do Spotify também possuem um comportamento semelhante em outras plat


X = df[['in_spotify_charts']]  # variável independente
y = df['in_deezer_charts']  # variável dependente


modelo = LinearRegression()
modelo.fit(X, y)

# Previsões
y_pred = modelo.predict(X)


print("Inclinação (coeficiente):", modelo.coef_[0])
print("Intercepto:", modelo.intercept_)
print("R² (coeficiente de determinação):", r2_score(y, y_pred))
```

```
Inclinação (coeficiente): 0.18520070945097256
Intercepto: 0.40843153879677674
R² (coeficiente de determinação): 0.36632253607225107
```

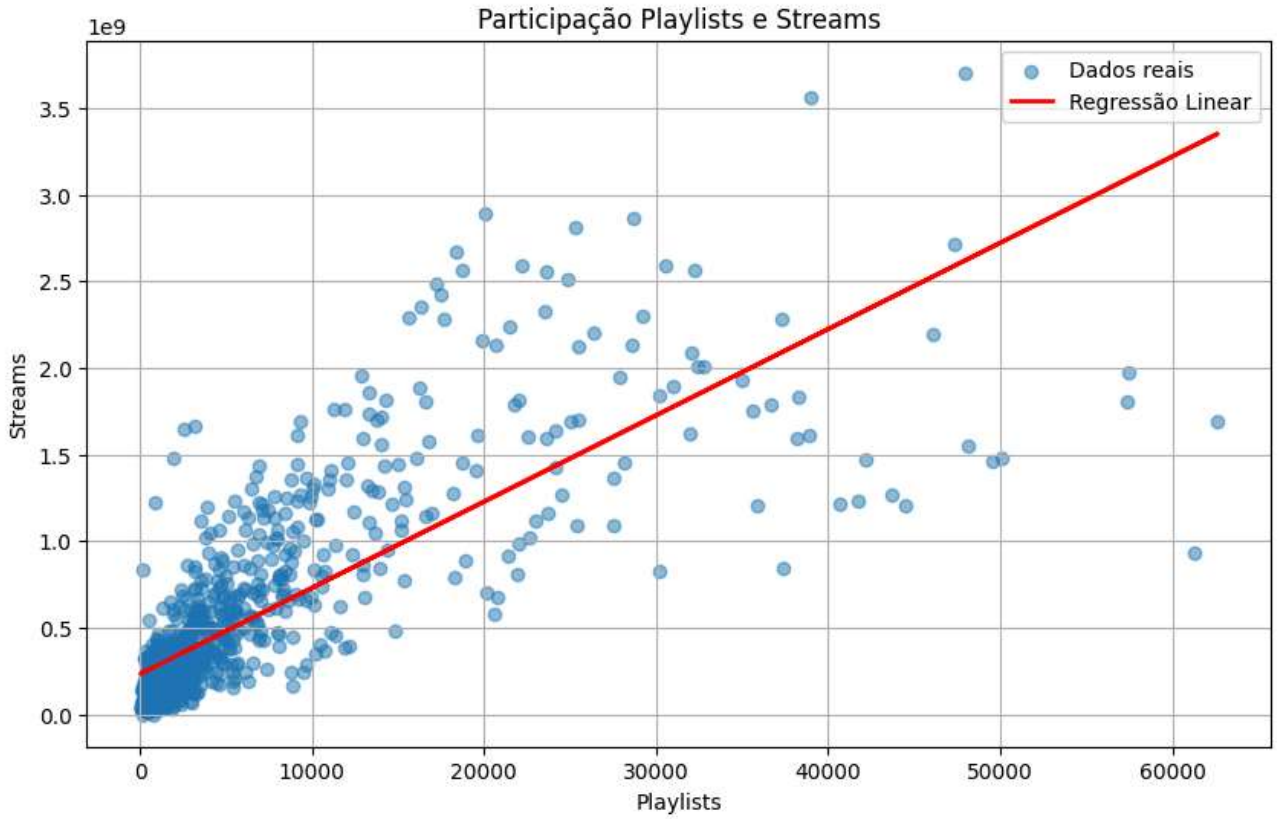
```
plt.figure(figsize=(10,6))
plt.scatter(X, y, alpha=0.6, label='Dados reais')
plt.plot(X, y_pred, color='red', label='Regressão linear')
plt.xlabel('Ranking Spotify')
plt.ylabel('Ranking Deezer')
plt.title('Correlação entre popularidade no Spotify e Deezer')
plt.legend()
plt.grid(True)
plt.show()
```




```
# Resultado: hipótese verdadeira.


# Hipótese: "A presença de uma música em um maior número de playlists está relacionada com um maior número de streams"


X = df[['total_playlists']]  # variável independente
y = df['streams_limpo']  # variável dependente


modelo = LinearRegression()
modelo.fit(X, y)

# Previsões
y_pred = modelo.predict(X)


print("Inclinação (coeficiente):", modelo.coef_[0])
print("Intercepto:", modelo.intercept_)
print("R² (coeficiente de determinação):", r2_score(y, y_pred))
```

```
Inclinação (coeficiente): 49776.738080397285
Intercepto: 232578557.7751062
R² (coeficiente de determinação): 0.6135391459920698
```

```
plt.figure(figsize=(10,6))
plt.scatter(X, y, alpha=0.5, label='Dados reais')
plt.plot(X, y_pred, color='red', linewidth=2, label='Regressão Linear')
plt.title('Participação Playlists e Streams')
plt.xlabel('Playlists')
plt.ylabel('Streams')
plt.legend()
plt.grid(True)
plt.show()
```




```
# Resultado: hipótese verdadeira.


# Hipótese: "Artistas com maior número de músicas no Spotify têm mais streams"


df = pd.read_csv('artistas_streams_musica.csv')  # Substitua pelo nome do seu arquivo
df.head()
```

	artista_corrigida	total_streams	quantidade_musica
0	Yuridia, Angela Aguilar	236857112	1
1	KALUSH	53729194	1
2	Ckay, AX'EL, Dj Yo	540539717	1
3	Karol G, Becky G	716591492	1
4	Dave	229473310	1

Próximas etapas: Gerar código com df | Ver gráficos recomendados | New interactive sheet

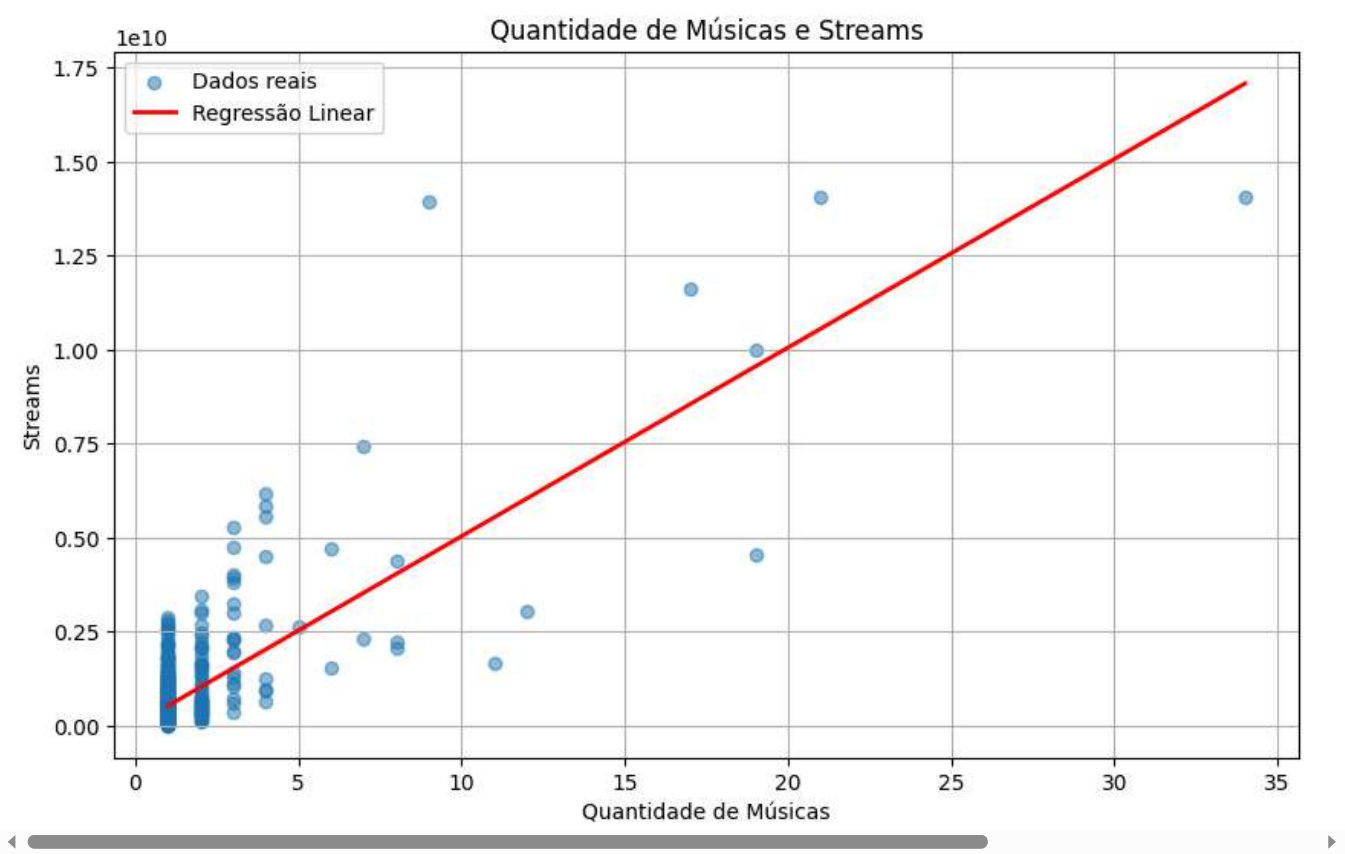
```
print(df.columns)
```

```
Index(['artista_corrigida', 'total_streams', 'quantidade_musica'], dtype='object')
```

```
X = df[['quantidade_musica']]  # variável independente
y = df['total_streams']  # variável dependente


modelo = LinearRegression()
modelo.fit(X, y)

# Previsões
y_pred = modelo.predict(X)


print("Inclinação (coeficiente):", modelo.coef_[0])
print("Intercepto:", modelo.intercept_)
print("R² (coeficiente de determinação):", r2_score(y, y_pred))
```

```
Inclinação (coeficiente): 501472940.58116406
Intercepto: 19043586.7578516
R² (coeficiente de determinação): 0.6067545162808194
```

```
plt.figure(figsize=(10,6))
plt.scatter(X, y, alpha=0.5, label='Dados reais')
plt.plot(X, y_pred, color='red', linewidth=2, label='Regressão Linear')
plt.title('Quantidade de Músicas e Streams')
plt.xlabel('Quantidade de Músicas')
plt.ylabel('Streams')
plt.legend()
plt.grid(True)
plt.show()
```




```
# Resultado: hipótese verdadeira.
```

```
# Hipótese: "As características da música influenciam o sucesso em termos de streams no Spotify"
```

```
df = pd.read_csv('tabela_caracteristicas.csv')  # Substitua pelo nome do seu arquivo
df.head()
```

	track_corrigida	Caracteristica	Valor	streams_limpo
0	Que Vuelvas	instrumentalness_%	0	2762
1	Que Vuelvas	acousticness_%	19	2762
2	Que Vuelvas	danceability_%	49	2762
3	Que Vuelvas	liveness_%	11	2762
4	Que Vuelvas	energy_%	64	2762

Próximas etapas: Gerar código com df | Ver gráficos recomendados | New interactive sheet

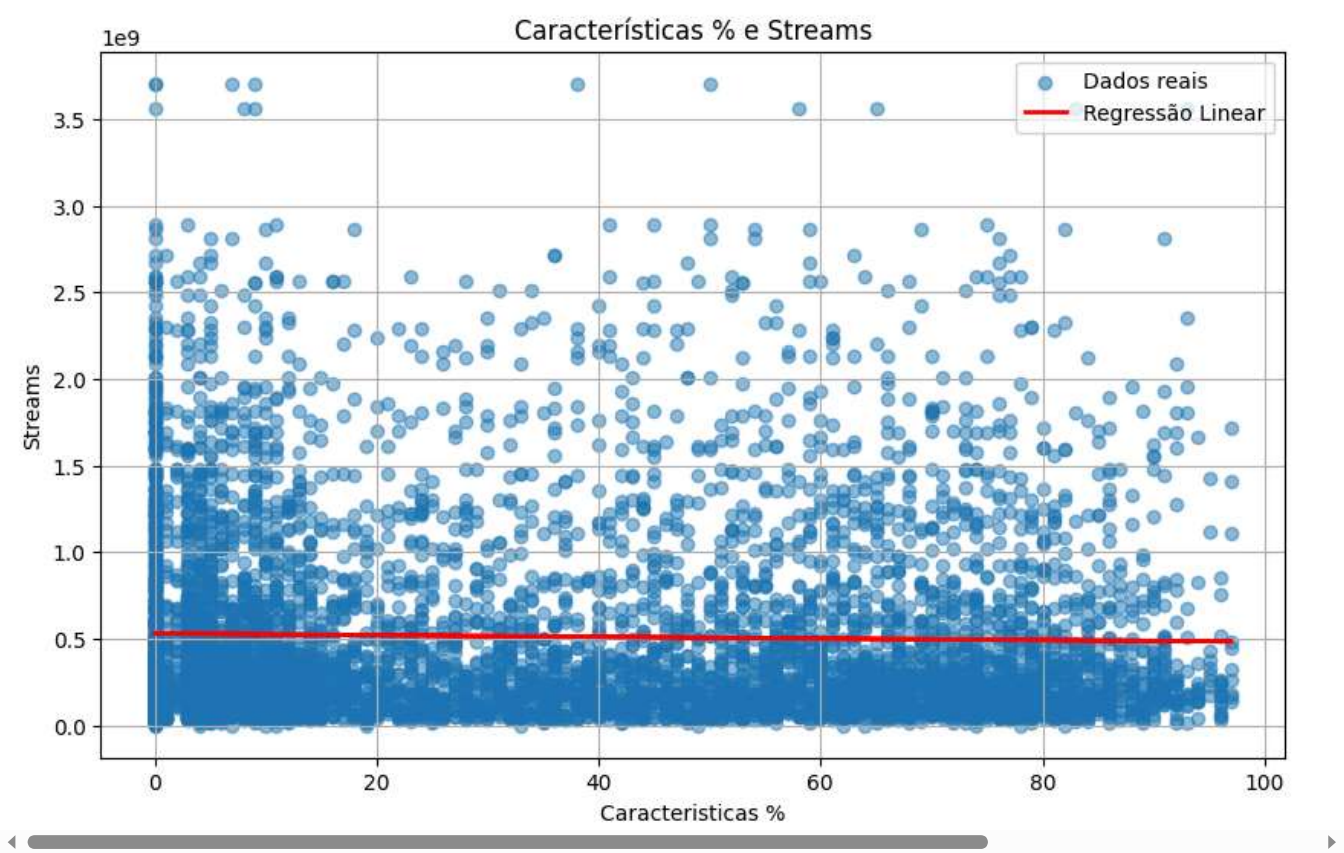
```
print(df.columns)
```

```
Index(['track_corrigida', 'Caracteristica', 'Valor', 'streams_limpo'], dtype='object')
```

```
X = df[['Valor']]  # variável independente
y = df['streams_limpo']  # variável dependente


modelo = LinearRegression()
modelo.fit(X, y)

# Previsões
y_pred = modelo.predict(X)


print("Inclinação (coeficiente):", modelo.coef_[0])
print("Intercepto:", modelo.intercept_)
print("R² (coeficiente de determinação):", r2_score(y, y_pred))
```

```
Inclinação (coeficiente): -472578.6682857088
Intercepto: 530588563.52554077
R² (coeficiente de determinação): 0.0006238081128357997
```

```
plt.figure(figsize=(10,6))
plt.scatter(X, y, alpha=0.5, label='Dados reais')
plt.plot(X, y_pred, color='red', linewidth=2, label='Regressão Linear')
plt.title('Características % e Streams')
plt.xlabel('Caracteristicas %')
plt.ylabel('Streams')
plt.legend()
plt.grid(True)
plt.show()
```




```
# Resultado: hipótese falsa, as características não influenciam positivamente o número de streams.
```